Using formal methods for automatic platform-independent code generation of run-time management

Mohammadsadegh Dalvandi, Asieh Salehi Fathabadi and Michael Butler
{md5g11, asf08r, mjb}@ecs.soton.ac.uk
University of Southampton

Run-Time Management (RTM) systems are used in embedded systems to dynamically adapt hardware performance to minimise energy consumption. A significant challenge is that the RTM software can require laborious manual adjustment across different hardware platforms due to the diversity of architecture characteristics. Model-driven development offers the potential to simplify the management of platform diversity by shifting the focus away from hand-written platform-specific code to platform-independent models from which platform-specific implementations are automatically generated. In this demonstration, we present a method for automatic generation of RTM implementations from platform-independent formal models. The methodology in designing the RTM systems uses a high-level mathematical language, Event-B, which can describe systems at different abstraction levels. A RTM software can be specified at a high level of abstraction in the first step. The Event-B refinement technique then can be applied in a stepwise manner to add more details to the initial abstract model to provide more functional details of the RTM. When the model is refined to a concrete level, a code generation tool is used to translate platform-independent Event-B RTM models to platform-specific implementations in C. Formal verification is used to ensure correctness of the Event-B models. The generated RTM code has been integrated with the PRiME[1] framework. The PRiME Framework provides services for managing cross layer connections between RTM algorithms, hardware devices and applications. The portability offered by our methodology is demonstrated by modelling a Reinforcement Learning (RL) based RTM and generating implementations for two different platforms (ARM Cortex- A7 and A15) that all achieve energy savings on the respective platforms. Figure 1 illustrates our platform-independent code generation approach.

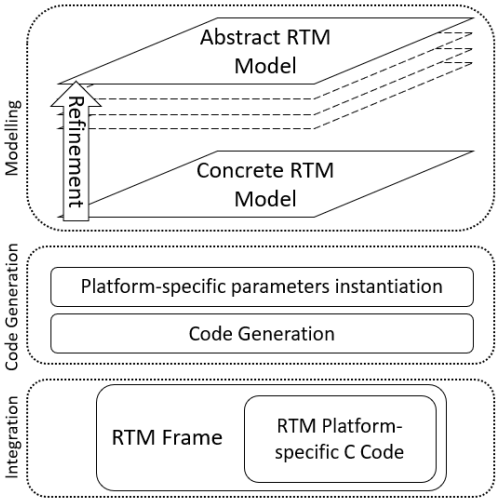

Figure 1 platform-independent code generation approach.

In the modelling level an abstract specification of the RTM is provided and then the model is enriched through several successive refinement steps. Our code generation tool then instantiates the model with platform-specific parameters and generates the platform-specific code from the instantiated model. Finally, the integration of the generated code with the PRiME framework is facilitated through a manually produced code called *RTM frame*.

Bibliography

Fathabadi, Asieh Salehi, et al. "**A model-based framework for software portability and verification in embedded power management systems.**" Journal of Systems Architecture 82 (2018): 12-23.

Acknowledgment

This work was supported by the PRiME programme grant EP/K034448/1 (http://www.primeproject.org).

[1] PRiME: Power-efficient, Reliable, Many-core Embedded systems